Reachability Solution Characterization of Parametric Real-Time Systems

Farn Wang and Hsu-Chun Yen

Dept. of Electrical Engineering, National Taiwan University
Taipei, Taiwan 106, Republic of China
{farn, yen}@cc.ee.ntu.edu.tw

Abstract

We investigate the problem of characterizing the solution spaces for timed automata augmented by unknown timing parameters (called *timing parameter automata* (TPA)). The main contribution of this paper is that we identify three non-trivial subclasses of TPAs, namely, *upper-bound, lower-bound* and *bipartite* TPAs, and analyze how hard it is to characterize the solution spaces. As it turns out, we are able to give complexity bounds for the sizes of the minimal (resp., maximal) elements which completely characterize the upward-closed (resp., downward-closed) solution spaces of upper-bound (resp., lower-bound) TPAs. For bipartite TPAs, it is shown that their solution spaces are not semilinear in general. We also extend our analysis to TPAs equipped with counters without zero-test capabilities.

1 Introduction

Timed automata have been a popular model in the research of formal description and verification of real-time systems [3]. In real-world applications, systems are usually described with unknown parameters to be analyzed. Here we use the term *timing parameters* to refer to those parameters which are compared with clocks in either timed automata [5] or parametric TCTL formulae [14–16]. A timed automaton extended with unknown timing parameters is called a *timing parameter automaton (TPA)*. A *valuation* of unknown parameters making the goal state reachable in a TPA is called a *solution*. In this paper, we are mainly concerned with the following problem:

> The *reachability solution characterization (RSC)* problem: Given a TPA A and a goal predicate η, formulate a representation for the solution space of A with respect to η.

By 'formulating a representation' we mean finding a proper characterization for the solution space so as to allow queries arisen frequently in verification (such as emptiness, membership, etc) to be answered effectively.

In [5], it has been shown that the emptiness problem (i.e., the problem of deciding whether there exists a parameter valuation under which the associated timed language is nonempty) becomes undecidable when three or more clocks are compared with unknown parameters in TPAs. Knowing such a limitation, a line of subsequent research has been focused on the RSC problem for a number of restricted versions of TPAs (see, e.g., [4,8,14–16]). These positive results obtained in the last few years have all been focused on unknown timing parameters in the specification of logic formulae. But in practice, it is more likely that design engineers will use unknown parameters in the system behaviour descriptions. Moreover, design engineers will be more interested in knowing the condition for solution parameters valuations than in knowing whether there exists a solution parameter valuation. In this work, we identify three subclasses of TPAs and investigate the complexity issue of their RSC problems. The three subclasses are called *upper-bound TPAs*, *lower-bound TPAs*, and *bipartite TPAs*. Consider a TPA and w.l.o.g., we assume that only $\leq$ and $<$ are used in the predicates of the TPA. An *upper-bound parameter* θ is one that only appears to the right of an inequality operator (e.g., $x < \theta, x \leq \theta$), whereas a *lower-bound parameter* θ appears to the left of an inequality operator (e.g., $\theta < x, \theta \leq x$). *Upper-bound* (resp. *lower-bound*) TPAs are those whose unknown parameters are all *upper-bound* (resp. *lower-bound*) parameters. *Bipartite* TPAs refer to those for which every unknown parameter is either a lower-bound parameter or an upper-bound parameter, but not both. Bipartite TPAs were considered in a recent article [10] in which the emptiness problem (undecidable for general TPA [5]) was shown to be decidable for such automata. In our setting, unknown parameters range over the set of natural numbers. As the work of [1] shows, unknown parameters of integer values can be used for modelling, for instance, the maximal number of retransmissions in the *Bounded Retransmission Protocol* (BRP), which is a data link protocol used by Philips. The interested reader is referred to [5,6,10] for TPAs with their parameters ranging over the set of real numbers. (Note that integer parameters are also considered in [5,6].)

Intuitively, what makes *upper-bound* (resp. *lower-bound*) TPAs easier to analyze, in comparison with their general counterparts, lies in the fact that for each of such TPAs, the solution space is *upward-closed* (resp. *downward-closed*). (A set S over k-dimensional vectors of natural numbers, for some k, is called *upward-closed* (resp., *downward-closed*) if $\forall u \in S, v \geq u \Longrightarrow v \in S$ (resp., $\forall u \in S, v \leq u \Longrightarrow v \in S$)). It is well known that an upward-closed set (resp., downward-closed set) is completely characterized by its *minimal* (resp., *maximal*) elements, which always form a finite set although the set might not be effectively computable in general. As we shall see later in this paper, we are able to give a complexity bound for the sizes of the minimal elements for a given upper-bound TPA. Our analysis is carried out in a way similar to a strategy proposed in [13] (by Valk and Jantzen), in which a sufficient and necessary condition was derived under which the set of minimal elements of an upward-closed set is guaranteed to be effectively computable. (Note, however, that [13] reveals no complexity bounds for the sizes of the minimal elements.) Taking advantage of certain properties offered by timed automata, we are able to refine Valk and Jantzen's approach to yield complexity bounds for the sizes of the minimal elements for the upward-closed sets associated with upper-bound TPAs, allowing us to characterize their solution spaces. This in turn answers the *RSC* problem for upper-bound TPAs. To a certain extent, our result supplements the work of [10] (in which the emptiness problem was shown to be decidable for bipartite TPAs) by tackling a more general problem. We are also able to extend our analysis to the model of upper-bound *timing parameter vector addition systems with states (TPVASSs)*, each of which can be viewed as a TPA equipped with counters without zero-test capabilities. Once the sizes of minimal elements become available, finding all such elements can be done by exhaustive search using the region graph technique, although it would clearly be interesting to develop smarter (and more efficient) algorithms. Some complexity results are also derived for lower-bound TPAs. For bipartite TPAs, we are able to show that their solution spaces are not semilinear in general, in spite of the fact that the emptiness problem is decidable [10].

We feel that the method developed in this paper for analyzing upward-closed sets is interesting in its own right. Our strategy provides a refinement over the approach proposed in [13] in the sense that the sizes of the minimal elements can now be deduced, provided that certain conditions are met. It would be interesting to seek additional applications of our technique.

2 Models of Parametric Timed Systems

Let Z (N and R^+, resp.) be the set of all integers (nonnegative integers, and nonnegative reals, resp.), and Z^k (N^k, resp.) be the set of k-dimensional vectors of integers (nonnegative integers, resp.). Let $\mathbf{0}$ be the *zero vector*. Let $v(i), 1 \leq i \leq k$, denote the i-th component of a k-dimensional vector v. Given two vectors u and $v (\in N^k)$, $u \leq v$ if $\forall 1 \leq i \leq k, u(i) \leq v(i)$, and $u < v$ if $u \leq v$ and $u \neq v$. We define the *norm* of v, denoted by $||v||$, to be $max\{|v(i)| \mid 1 \leq i \leq k\}$, i.e., the absolute value of the largest component in v. For a set of vectors $V = \{v_1, ..., v_m\}$, the *norm* of V is defined to be $max\{||v_i|| \mid 1 \leq i \leq m\}$. In our subsequent discussion, we let $N_\infty = N \cup \{\infty\}$ (∞ is a new element capturing the notion of something being 'arbitrarily large', and for every $t \in N, t < \infty$ holds). We also let $N^k_\infty = (N \cup \{\infty\})^k = \{(v_1, ..., v_k) \mid v_i \in (N \cup \{\infty\}), 1 \leq i \leq k\}$. For a $v \in N^k_\infty$, we also write $||v||$ to denote $max\{v(i) \mid v(i) \neq \infty\}$, (i.e., the largest component in v excluding ∞) if $v \neq (\infty, ..., \infty)$; $||(\infty, ..., \infty)|| = 1$. Unless stated otherwise, we always assume that numbers are represented in *binary*, and the *size* of a number $t \in N$ is $\lceil log_2 t \rceil$.

A set $U (\subseteq N^k)$ is called *upward-closed* if $\forall u \in U$, $\forall v, v \geq u \Longrightarrow v \in U$. An element u $(\in N^k)$ is said to be *minimal* if there is no v $(\neq u)$ $\in U$ such that $v < u$. We write $min(U)$ to denote the set of minimal elements of U. For an element $v \in N^k_\infty$, let $reg(v) = \{w \in N^k \mid w \leq v\}$. A set $D (\subseteq N^k)$ is called *downward-closed* if $\forall u \in D$, $\forall v, \mathbf{0} \leq v \leq u \Longrightarrow v \in D$. An element u $(\in N^k_\infty)$ is said to be *maximal* if there is no v $(\neq u)$ $\in D$ such that $v > u$. We write $max(D)$ to denote the set of maximal elements of D. For a given dimension, it is well known that every upward-closed (resp., downward-closed) set has a finite number of minimal (resp., maximal) elements. However, such finite sets may not be effectively computable in general. In an article [13] by Valk and Jantzen, the following result was proven which suggests a sufficient and necessary condition under which the set of minimal elements of an upward-closed set is effectively computable:

THEOREM 1 *([13]) For each upward-closed set $U (\subseteq N^k)$, $min(U)$ is effectively computable iff for every $v \in N^k_\infty$, the problem 'reg(v) $\cap U \neq \emptyset$?' is decidable.*

2.1 Timing parameter automata (TPA)

Given a set P of basic propositions, a set X of clocks, and a set H of unknown parameters, a *state predicate* η of P, X, and H has the following syntax rules.

$$\eta ::= \mathit{false} \mid p \mid x \sim c \mid x \sim \theta \mid \eta_1 \vee \eta_2 \mid \neg \eta_1$$

where $p \in P$, $x \in X$, $c \in N$, $\theta \in H$, $\sim \in \{\leq, <, =, \geq, >\}$, and η_1, η_2 are state predicates. Notationally, we let $B(P, X, H)$ be the set of all state predicates on P, X, and H. Parentheses and traditional shorthands like $\Rightarrow, \wedge$ can also be used. It is worthy of pointing out that, like the model given in [3], clock constraints are assumed to be *diagonal-free* (i.e., the comparison between two clocks is not allowed). The interested reader is referred to [7] for details about why relaxing the diagonal-free constraints will render several forward analysis algorithms reported in the literature incorrect.

Definition 1 (Timing parameter automata, state and interpretation) A *timing parameter automaton* (TPA) A is a tuple $(Q, q_0, X, H, \mu, E, \tau, \pi)$, where Q is a finite set of modes (operation modes, or control locations), $q_0 \in Q$ is the initial mode, X is a set of clocks with readings in R^+, H is a set of parameter variables with values in N, μ is a mapping from Q such that for each $q \in Q$, $\mu(q) \in B(\emptyset, X, H)$ is the invariance condition true in q, $E \subseteq Q \times Q$ is the set of transitions, $\tau : E \mapsto B(\emptyset, X, H)$ is a mapping which defines the transition-triggering conditions, and $\pi : E \mapsto 2^X$ defines the set of clocks to be reset on each transition. A *state* of TPA A is a pair (q, ν) such that $q \in Q$ and ν is a mapping from X to R^+ (i.e., ν represents the current clock readings). Let U_A be the state set of A. An *interpretation* I for H is a mapping from H to N. ■

Let A be specified in Definition 1. Given an interpretation I, A^I is the timed automaton obtained from A with all parameters interpreted according to I. Given a predicate $\eta \in B(P, X, H)$ and an interpretation I, η^I is the new predicate obtained from η with all parameters interpreted according to I.

Definition 2 (Satisfaction of state predicate with interpretation) A state (q, ν) *satisfies* state predicate $\eta \in B(Q, X, H)$ with interpretation I, written as $(q, \nu) \models_I \eta$, iff

- $(q, \nu) \not\models_I$ *false*;
- $(q, \nu) \models_I q'$ iff $q = q'$;
- $(q, \nu) \models_I x \sim \theta$ iff $\nu(x) \sim I(\theta)$ where $\theta \in H$;
- $(q, \nu) \models_I x \sim c$ iff $\nu(x) \sim c$ where $c \in N$;
- $(q, \nu) \models_I \eta_1 \vee \eta_2$ iff $(q, \nu) \models_I \eta_1$ or $(q, \nu) \models_I \eta_2$; and
- $(q, \nu) \models_I \neg\eta_1$ iff $(q, \nu) \not\models_I \eta_1$.

If for all I, we have $(q, \nu) \models_I \eta$, then we may write $(q, \nu) \models \eta$. ■

Definition 3 (Transitions) Given two states $(q, \nu), (q', \nu')$, there is a *mode transition* from (q, ν) to (q', ν') in A with interpretation I, in symbols $(q, \nu) \rightarrow_I (q', \nu')$, iff, $(q, q') \in E$, $(q, \nu) \models_I \mu(q) \wedge \tau(q, q')$, $(q', \nu') \models_I \mu(q')$, $\forall x \in \pi(q, q')(\nu'(x) = 0)$, and $\forall x \notin \pi(q, q')(\nu'(x) = \nu(x))$. ■

For notational convenience, given a clock reading ν and a $\delta \in R^+$, we define a new clock reading $\nu + \delta$ to be $(\nu + \delta)(x) = \nu(x) + \delta$, for all $x \in X$. We also write $(q, \nu) + \delta$ to denote the new state $(q, \nu + \delta)$.

Definition 4 ((q, ν)-run of interpreted TPA) An infinite computation of A starting at state (q, ν) with interpretation I is called a (q, ν)-*run* and is a sequence $((q_1, \nu_1, t_1), (q_2, \nu_2, t_2), \ldots)$ such that

- $q = q_1$ and $\nu = \nu_1$;
- $t_1 t_2 \ldots$ is a monotonically increasing sequence such that for each $t \in R^+$, there is an $i \in N$ with $t_i \geq t$ (meaning that the run is divergent);
- for each integer $i \geq 1$ and for each real $0 \leq \delta \leq t_{i+1} - t_i$, $(q_i, \nu_i) + \delta \models_I \mu(q_i)$ (meaning that the invariance condition $\mu(q_i)$ continuously holds throughout the time interval $[t_i, t_{i+1}]$); and

- for each $i \geq 1$, A goes from (q_i, ν_i) to (q_{i+1}, ν_{i+1}) because of
  - a mode transition, i.e., $t_i = t_{i+1} \wedge (q_i, \nu_i) \rightarrow_I (q_{i+1}, \nu_{i+1})$; or
  - time passage, i.e., $t_i < t_{i+1} \wedge (q_i, \nu_i) + t_{i+1} - t_i = (q_{i+1}, \nu_{i+1})$. ■

2.2 The reachability solution characterization problem

Let $\langle 0 \rangle$ be the mapping that maps every clock to zero. The initial state of a TPA A is $(q_0, \langle 0 \rangle)$. Given a TPA A, a goal state-predicate $\eta \in B(Q, X, H)$, and an interpretation I, we say that η is *reachable* in A with I, in symbols $A \rightsquigarrow_I \eta$, iff there exist a $(q_0, \langle 0 \rangle)$-run $= ((q_1, \nu_1, t_1), (q_2, \nu_2, t_2), \ldots)$ in A, an $i \geq 1$, and a $\delta \in [0, t_{i+1} - t_i]$, such that $(q_i, \nu_i) + \delta \models_I \eta$. An interpretation I satisfying $A \rightsquigarrow_I \eta$ is called a *solution* for A and η. The set of all solutions forms the so-called *solution space*. With respect to a given pair of A and $\eta \in B(Q, X, H)$, the problem of *finding a proper characterization for the solution space of A with respect to η* arises naturally in many real-world applications. Such a problem is called the *Reachability Solution Characterization (RSC)* problem. Throughout the rest of this paper, we write $RSC(A, \eta)$ to denote the solution space of TPA A with respect to predicate η.

2.3 Lower-bound, upper-bound, and bipartite TPAs

One of the major motivations in this work is to find practical classes of TPAs for which we can develop algorithms with known complexities for their RSC problem. First, we need the following concepts. A predicate $\eta \in B(P, X, H)$ is in *literal form* iff in η, negation symbols only appear in front of elements in P; there is no negative signs immediately before inequality literals; and only $\leq$ and $<$ are used in inequality atoms. Every predicate can be transformed to a literal form in linear time. (For instance, $\neg(x \geq \theta)$ has $x < \theta$ as its equivalent literal form.) A TPA $A = (Q, q_0, X, H, \mu, E, \tau, \pi)$ is called a *literal TPA* iff $\mu(q)$ is in literal form for all $q \in Q$; and $\tau(q, q')$ is also in literal form for all $q, q' \in Q$. Notice that every TPA can also be transformed to a literal TPA in linear time. In a literal TPA, if an unknown parameter θ appears to the right of an inequality operator in a literal (e.g. $x \leq \theta$, $x < \theta$), then θ is called an *upper-bound parameter*. If it appears to the left of an inequality operator in a literal (e.g. $\theta \leq x$, $\theta < x$), then it is called a *lower-bound parameter*.

Definition 5 *(***Bipartite, lower-bound, and upper-bound TPAs***)* A *bipartite* TPA A is a literal TPA such that its set $\underline{H}$ of lower-bound parameters and set $\overline{H}$ of upper-bound parameters are disjoint, i.e. $\underline{H} \cap \overline{H} = \emptyset$. If $\underline{H} = \emptyset$, then A is also called an *upper-bound TPA*. If $\overline{H} = \emptyset$, then A is also called a *lower-bound TPA*. A predicate η in literal form is called an *upper-bound* (resp., *lower-bound*) predicate if all of its constituent parameters are *upper-bound* (resp., *lower-bound*) parameters. ■

There are two interpretations on a bipartite TPA which are important in defining the computability of the RSC problem. The first is the *maximum interpretation* I^M with which $I^M(\underline{\theta}) = 0$ for all $\underline{\theta} \in \underline{H}$; and $I^M(\overline{\theta}) = \infty$ for all $\overline{\theta} \in \overline{H}$. The second is the *minimum interpretation* I^m with which $I^m(\underline{\theta}) = \infty$ for all $\underline{\theta} \in \underline{H}$; and $I^m(\overline{\theta}) = 0$ for all $\overline{\theta} \in \overline{H}$. Note that maximum and minimum interpretations are not really interpretations as we defined in Definition 1 which does not map parameters to ∞. While interpreting $(q, \nu) \models_{I^M} \eta$ and $(q, \nu) \models_{I^m} \eta$, we shall assume $c \sim \infty = \mathit{true}$ and $\infty \sim c = \mathit{false}$, where $\sim \in \{<, \leq\}$.

We assume the basic knowledge of *region graph* constructions for timed automata presented in [2,3]. Suppose that the biggest timing constant used in A and η is $C_{A:\eta}$. In [2,3], given a timed automaton A (or a TPA with an interpretation), a *region* for a state (q, ν) is a triple (q, γ, ϕ) such that γ records the integer parts of clock readings, at (q, ν), up to $C_{A:\eta}$ (when a clock reading is bigger than $C_{A:\eta}$, it is represented as ∞), and ϕ records the total ordering of the fractional parts of zero and clock readings at (q, ν). As [2,3] indicates, the reachability problem of timed automata can be solved in the domain of *region graphs*, each of which has its region set as the node set and (timed and discrete) transition relation from regions to regions as the arc set. A rough bound on the number of regions in a region graph for A^{I_M} was computed as $\Lambda(|Q|, C_{A:\eta}, |X|) = 2|Q|((2 + C_{A:\eta})|X|)^{|X|}$. Here $|X|^{|X|}$ is a rough bound w.r.t. the ϕ-component. Coefficient 2 at the beginning indicates that for each total-ordering among the fractional parts of clock readings, either all such fractional parts are nonzero or the first one is zero. Accordingly, there are two kinds of regions. The first is for regions in which some clocks are of an integer reading within $C_{A:\eta}$, whereas the second is for those in which no clock is. In the case that the fractional parts of clocks' readings are the same, then along each region paths of time progression, the regions will alternate through between these two kinds of regions. But from a region of the first kind to one of the second, the elapsed time along the path does not increment by one. In fact, it takes two alternations in sequence, at least, to increment the elapsed time by one. Thus we can divide the bound (on numbers of regions) by two, and get a tighter bound on the elapsed times along paths.

LEMMA 2 *$A \rightsquigarrow_{I^M} \eta$ iff there is a run, of less than $(\Lambda(|Q|, C_{A:\eta}, |X|) - 1)/2$ time units long, from the initial state $(q_0, \langle 0 \rangle)$ to a state satisfying η.*

From now on, we shall let $\Gamma_{A:\eta} = (\Lambda(|Q|, C_{A:\eta}, |X|) - 1)/2$ for convenience. In subsequent sections, we consider the problem of characterizing $RSC(A, \eta)$ when both TPA A and goal predicate η are upper-bound (or lower-bound). It is worthy of noting that in either case, the clock constraints in η can be built into TPA A. Consequently, the RSC problem can further be modified into one with only state reachability.

3 Computing Minimal Elements for Models with Upward-Closed Solution Spaces

3.1 Upper-bound TPAs

Now consider upper-bound TPAs with upper-bound goal predicates. By establishing an ordering on the elements of H (i.e., $H = \{\theta_1, ..., \theta_k\}$, for some k), an interpretation for parameters in H can now be regarded as a $k-$dimensional vector in N^k. With a slight abuse of notation, for an interpretation I we write $I(\theta)$ to denote $I(i)$, where $\theta = \theta_i$. Given an interpretation I and a $\Delta \geq \mathbf{0}(\Delta \in N^k)$, we define $I + \Delta$ as the new interpretation such that for all $\theta \in H$, $(I + \Delta)(\theta) = I(\theta) + \Delta(\theta)$. The following lemma shows that the solution space for each upper-bound TPA w.r.t. an upper-bound goal predicate is *upward-closed*.

LEMMA 3 *For any upper-bound TPA A and upper-bound goal predicate η, if $A \rightsquigarrow_I \eta$ is true, then $\forall \Delta \geq \mathbf{0}$, $(A \rightsquigarrow_{I+\Delta} \eta)$. In words, the set of interpretations satisfying η is upward-closed.*

In view of the above lemma, each solution I can actually be regarded a *representative* for a convex space of solutions, called *funnel* of I. Given an interpretation I, we use $\langle I \rangle$ to represent the funnel pointing at I, i.e., $\langle I \rangle = \{I + \Delta \mid \Delta \geq \mathbf{0}\}$. I is called the *point* of funnel $\langle I \rangle$. (Note that $\langle I \rangle$ has a unique minimal element, namely, I.) A set of funnels $\langle I_1 \rangle, \ldots, \langle I_m \rangle$ is called *mutually independent* iff each funnel is not a subset of the unions of the others, that is, $\forall 1 \leq i \leq m(\langle I_i \rangle \not\subseteq \bigcup_{1 \leq j \leq m; i \neq j} \langle I_j \rangle)$, or equivalently $\forall 1 \leq i < j \leq m\ \exists \alpha \in H\ \exists \alpha' \in H(I_i(\alpha) < I_j(\alpha) \wedge I_i(\alpha') > I_j(\alpha'))$.

Given an upper-bound TPA and an η, Lemma 3 suggests that $RSC(A, \eta)$ is upward-closed. Using the basic theory of timed automata (see, e.g., [3]), the problem of, given an interpretation $I \in N_\infty^k$, deciding '$reg(I) \cap RSC(A, \eta) \neq \emptyset$?' is clearly decidable. This observation, in conjunction with Theorem 1, yields the computability of the set of minimal elements of $RSC(A, \eta)$, although it reveals no information regarding the size of $min(RSC(A, \eta))$. In the remainder of this section, we shall take advantage of certain properties of timed automata to derive complexity bounds for computing $min(RSC(A, \eta))$. Our analysis involves the following two steps. We first show that the solution space $RSC(A, \eta)$ is a finite union of funnels. Then, with an inductive scheme on the number of unknown upper-bound parameters, we derive a finite bound on the magnitudes of parameter values of point solutions of the funnels in the finite union. The position of an existent solution is important in identifying the finite structure of the solution space. Let I^a be the interpretation that maps every $\theta \in H$ to $1+a$. Lemma 2 implies that if the solution space for an upper-bound TPA is nonempty, then $I^{(\Lambda(|Q|,C_{A:\eta},|X|)-1)/2}$ is a solution.

Let J be a *partial interpretation* of the parameters in H, that is, J is undefined for some parameters in H. For convenience, we write $J(\alpha)=\infty$, if α is undefined in J. (By doing so, J becomes a vector in N_∞^k, where $k=|H|$.) We conveniently use $\langle J\rangle$ as the union of all $\langle I\rangle$ such that $I(\alpha)=J(\alpha)$ for every α defined in J (i.e., $J(\alpha)\neq\infty$). Notice that $\langle J\rangle = \bigcup_{I\in N^k, \forall J(\alpha)\neq\infty(I(\alpha)=J(\alpha))}\langle I\rangle$ Thus we may also write $\langle I\rangle \subseteq \langle J\rangle$ if I agrees with J on every α defined in J. Given a partial interpretation J, in symbols, we let $\bar{H}^J$ be the set of variables in H uninterpreted by J, that is $\bar{H}^J = \{\alpha \mid J(\alpha) \text{ is undefined}\}$. In $\langle J\rangle$, there can be non-solution interpretations for A and η. The following notation is for the characterization of those solution interpretations in $\langle J\rangle$. Given a J, we let $\Omega^J_{A:\eta}$ be the space of solutions I for $A \rightsquigarrow_I \eta$ with $\langle I\rangle \subseteq \langle J\rangle$. If J happens to be a total interpretation, then (1) $\Omega^J_{A:\eta} = \langle J\rangle = \{v \mid v \geq J\}$ in case $A \rightsquigarrow_J \eta$; and (2) $\Omega^J_{A:\eta}=\emptyset$ otherwise. For convenience, given a partial interpretation J and $a\in N$, we let $J[\alpha := a]$ be a new partial interpretation that agrees with J in every parameter except that $J[\alpha := a](\alpha) = a$.

LEMMA 4 *If J is a partial interpretation and $I \in \Omega^J_{A:\eta}$ is a total interpretation, then*

$$\Omega^J_{A:\eta} = \langle I\rangle \cup \bigcup_{\alpha\in\bar{H}^J;0\leq a<I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta} \tag{1}$$

PROOF. We prove this by an induction on the size of $\bar{H}^J$. In the base case, $|\bar{H}^J|=0$, J happens to be a total interpretation, and moreover $\bigcup_{\alpha\in\bar{H}^J;0\leq a<I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta}=\emptyset$. Thus the base case is proven.

In the induction step, we assume that the lemma is true for all $|\bar{H}^J|\leq k$ with $k\geq 0$. Let us consider a solution interpretation $I'\in\Omega^J_{A:\eta}$. If $\langle I'\rangle \subseteq \langle I\rangle$, then the lemma is proven. Now let us consider the case that $\langle I'\rangle \not\subseteq \langle I\rangle$. In this case, this means that there is an $\alpha\in\bar{H}^J$ such that $I'(\alpha)<I(\alpha)$. By enumerating all the possibilities of $\alpha\in\bar{H}^J$ and all the possible values in $[0, I(\alpha))$, we obtain the expression of $\bigcup_{\alpha\in\bar{H}^J;0\leq a<I(\alpha)} \Omega^{J[\alpha:=a]}_{A:\eta}$. Thus the induction step is proven. Thus the lemma is also proven. ■

The importance of Lemma 4 is that once we can find a solution interpretation I in the solution space, the lemma suggests a way to inductively and compositionally construct the solution space by means of unions of funnels. But according to Lemma 2, we do know how to find this special interpretation I based on a given partial interpretation. As we shall see later, the ability to effectively compute a total interpretation $I \in \Omega^J_{A,\eta}$ plays a critical role in deriving a complexity bound for the size of $min(RSC(A,\eta))$.

3.2 Complexity analysis

Let $(\infty, ..., \infty)$ be the partial interpretation that is undefined on every parameter. Lemma 4, together with the fact that $RSC(A,\eta) = \Omega^{(\infty,...,\infty)}_{A:\eta}$, suggests an algorithm for computing the constituent funnels of $RSC(A,\eta)$, provided that $I \in \Omega^J_{A:\eta}$ is computable for every partial interpretation J. That is, if we view $\Omega^J_{A:\eta}$ as a procedure-call with parameters A, η, and J, then we can construct the solution space representation by invoking $\Omega^{(\infty,...,\infty)}_{A:\eta}$.

By examining all components of Formula (1), we find that every component in (1) is with straightforwardly known complexity except $I \in \Omega^J_{A:\eta}$. It is obvious that if we can find bounds on the vector $I \in \Omega^J_{A:\eta}$ for each J, then we can multiply and sum up all the component complexities to derive the complexity for the RSC problem. The major difficulty is to carefully account for all the component complexities so that bounds can be derived. Let A^J be the new timed automaton obtained from A by substituting every defined θ in J for $J(\theta)$; and substituting every undefined θ' in J for ∞. The bounds can be obtained by using Lemma 2. That is, we can construct the region graph for A^J and η and use the length of the longest simple path in the graph to bound the vector components in I. In the same reasoning of Lemma 2, we know that there is an interpretation $I \in \Omega^J_{A:\eta}$ making $A \leadsto_I \eta$ iff $A^J \leadsto \eta$. According to [3], the size of region graph is bounded by

$$2|Q| \cdot |X|^{|X|} \cdot (C_{A^J:\eta} + 2)^{|X|} \tag{2}$$

In a region graph of this size, the time-span of the shortest path from one region to another can always be bounded by $|Q| \cdot |X|^{|X|} \cdot (C_{A^J:\eta} + 2)^{|X|}$. According to the same reasoning of Lemma 2, we can bound each component of the I with $|Q| \cdot |X|^{|X|} \cdot (C_{A^J:\eta} + 2)^{|X|}$. Notice that $C_{A^J:n}$ is now the biggest timing constant used in A with parameters replaced according to J. It should also be noted that when $J(\alpha) = \infty$, then $J(\alpha)$ is not considered as a timing constant candidate for $C_{A^J:n}$.

LEMMA 5 *For every partial interpretation J, there is an $I \in \Omega^J_{A:\eta}$ such that for every $\theta \in \bar{H}^J$, $I(\theta) \leq |Q| \cdot |X|^{|X|} \cdot (C_{A^J:\eta} + 2)^{|X|}$.*

We want to construct the inductive definition of the magnitude of $C_{A^J:\eta}$. That is, we want to define $C_{A^J:\eta}$ based on those partial interpretations which define one less parameters than J does. The following lemma unwinds Formula (2) for a bound on the complexity of $I \in \Omega^J_{A:\eta}$ for each J. For convenience, we let

$|J|$ be the number of parameters defined in J.

LEMMA 6 *In Formula (1), for every partial interpretation J, there is an $I \in \Omega^J_{A:\eta}$ such that for every $\theta \in \bar{H}^J$, $I(\theta)$ is $O((|Q| \cdot |X|^{|X|})^{\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+2)^{|X|^{1+|J|}})$.*

PROOF. Base case: $|J| = 0$. In this case, the bound is $|Q| \cdot |X|^{|X|} \cdot (C_{A:\eta}+2)^{|X|}$, which is exactly the time-span bound of the longest simple path in the region graph for A and η with I^M. This case is true according to Lemma 2.

The inductive hypothesis is that there is an $I \in \Omega^J_{A:\eta}$ such that for every $\theta \in \bar{H}^J$, $I(\theta)$ is $O\left(\left(|Q| \cdot |X|^{|X|}\right)^{\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+2)^{|X|^{1+|J|}}\right)$. This means that in the induction step, the biggest timing constant used in $A^{J[\theta:=a]}$ is of the same complexity. Thus we deduce, according to Formula (2), that the size bound of the region graph for $A^{J[\theta:=a]}$ is

$$\begin{aligned}
&O\left(2|Q| \cdot |X|^{|X|} \cdot \left(\left(|Q| \cdot (|X|^{|X|})\right)^{\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+2)^{|X|^{1+|J|}}+2\right)^{|X|}\right)\\
&= O\left(2|Q| \cdot |X|^{|X|} \cdot \left(\left(|Q| \cdot (|X|^{|X|})\right)^{|X|\sum_{0\leq i\leq |J|}|X|^i} \cdot (C_{A:\eta}+2)^{|X|^{1+|J|}\cdot |X|}\right)\right)\\
&= O\left(\left(|Q| \cdot |X|^{|X|}\right)^{\sum_{0\leq i\leq 1+|J|}|X|^i} \cdot (C_{A:\eta}+2)^{|X|^{2+|J|}}\right)
\end{aligned}$$

Since $|J|+1 = |J[\theta := a]|$ and the reachability does not need a path along which the elapsed time is greater than the number of regions, thus the complexity is proven. ■

Using Lemmas 4 and 6, we have the following result:

THEOREM 7 *Given an upper-bound TPA A and an upper-bound predicate η, $||min(RSC(A,\eta))||$ is bounded by $O((|Q| \cdot |X| \cdot C_{A:\eta})^{|X|^{c*|H|}})$, where c is a constant.*

Once the sizes of minimal elements become available, finding all such elements can be done by exhaustive search using the region graph technique, although it would clearly be desirable to develop smarter (and more efficient) algorithms.

3.3 An example

In Figure 1, we have a simple upper-bound TPA to show how our algorithm inductively invokes the basic region graph construction procedure in [2] to build the solution space characterization. There are two ways that we can

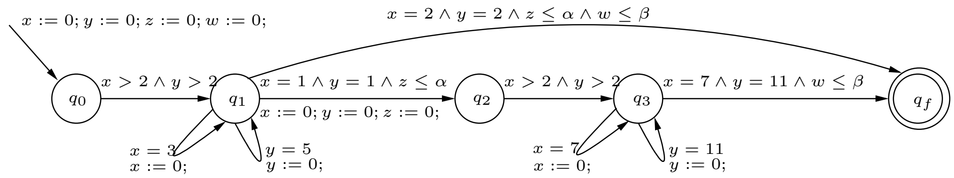

Fig. 1. An example of upper-bound TPAs.

reach the final mode q_f. The first is from mode q_0 through q_1 to q_f. The time elapsed in such a path is of the pattern $3 \cdot 5 \cdot i + 2$ for some $i \geq 1$. Note that arc (q_0, q_1) guarantees that $i \geq 1$. Since clocks z and w are never reset in this way, when q_f is reached, $z = w \geq 17$ and we can infer that $\alpha = \beta = 17$ characterizes a minimal solution.

The second way to reach q_f is from q_0 and through q_1, q_2, and q_3. The time elapsed in the computation from q_0 to q_2 is of the pattern $3 \cdot 5 \cdot i + 1$ for some $i \geq 1$. Thus in this way, we can infer that $\alpha \geq 16$ when transition (q_1, q_2) takes place. Then the time elapsed in the computation from q_2 to q_f is of the pattern $7 \cdot 11 \cdot j$ for some $j \geq 1$. Again, arc (q_2, q_3) guarantees that $j \geq 1$. Thus in this way, the time elapsed from q_0 through q_1, q_2, q_3 to q_f is of the pattern $15i + 1 + 77j$ for some $i \geq 1, j \geq 1$. From this, we can infer that $\alpha = 16$ and $\beta = 93$ characterizes another minimal solution.

In the execution of our algorithm, we will first find a minimal solution by interpreting $\alpha = \infty$ and $\beta = \infty$. Two candidates are $(17, 17)$ and $(16, 93)$. In formula (1) in Lemma 4, if we choose $I = (17, 17)$, then the formula says that the solution space is characterized by $\langle(17, 17)\rangle \cup (\bigcup_{0 \leq a < 17} \Omega_{A:\eta}^{(a,\infty)}) \cup (\bigcup_{0 \leq a < 17} \Omega_{A:\eta}^{(\infty,a)})$. The characterization of $\Omega_{A:\eta}^{(a,\infty)}$ can be obtained by interpreting α as a in A. From the reasoning in the last two paragraphs, we know that when $0 \leq \alpha < 17$, the only solution is $(16, 93)$. This can be obtained by finding the elapsed time of the path to q_f with $\alpha = 16$ and $\beta = \infty$. Similarly, the characterization of $\Omega_{A:\eta}^{(\infty,a)}$ can be obtained by interpreting β as a in A. From the reasoning above, we know that when $0 \leq \beta < 17$, there is no solution.

3.4 Timing parameter vector addition systems with states

Our technique can also be applied to analyzing upper-bound *timing parameter vector addition systems with states* (TPVASSs). An *m-dimensional vector addition system* (VAS) is a pair (v_0, V) where $v_0 \in N^m$ is called the *start vector*, and V, a finite subset of Z^m, is called the set of *addition rules*. A vector $z \in N^m$ is said to be reachable in VAS (v_0, V) if $z = v_0 + v_1 + \cdots + v_j$ for some $j \geq 0$, where each $v_i (1 \leq i \leq j)$ is in V and, for each $1 \leq i \leq j$, $v_0 + v_1 + \cdots + v_i \geq \mathbf{0}$. The *covering* problem of VASs is that of, given a VAS (v_0, V) and a vector v, deciding whether there is a reachable vector z such that

$z \geq v$ (i.e., z covers v). An *m-dimensional vector addition system with states* (VASS) is a 5-tuple (v_0, V, p_0, S, δ) where v_0 and V are the same as defined above, S is a finite set of *states*, $\delta \subseteq S \times S \times V$ is the *transition relation*, and $p_0 \in S$ is the *initial state*. Elements (p, q, v) of δ are called *transitions* and are usually written as $p \rightarrow (q, v)$. A configuration of a VASS is a pair (p, u) where $p \in S$ and $u \in N^m$. (p_0, v_0) is the *initial configuration*. The transition $p \rightarrow (q, v)$ can be applied to the configuration (p, u) and yields the configuration $(q, u + v)$, provided that $u + v \geq \mathbf{0}$. The reader is referred to [9,11] for more about VASs and VASSs.

A *timing parameter VASS* (TPVASS) A is a tuple $(v_0, V, Q, q_0, X, H, \mu, E, \tau, \pi)$ where v_0 and V represent the start vector and the addition rules, respectively, associated with TPA $(Q, q_0, X, H, \mu, E, \tau, \pi)$. One may view a TPVASS as a TPA equipped with counters without zero-test capabilities. The *dimension* of a TPVASS is the dimension of its constituent VAS. It is important to point out that the way *time* is introduced in this computational model differs from that in the conventional *timed* (or *time*) Petri nets. Unlike the case in the strong firing semantics of *timed* (or *time*) Petri nets, it is <u>not</u> required to fire all the enabled transitions at any point in time during the course of a computation. In our setting it is perfectly legal for time to elapse, causing enabled transitions to become disabled without being fired.

Using the technique of region graphs, we have:

LEMMA 8 *Given a TPVASS A, an interpretation I, and an upper-bound predicate $\eta \in B(Q, X, H)$, we can construct a VASS $M_{A,\eta,I} = (v_0, V, p_0, S, \delta)$ and a state $s \in S$ such that $A \leadsto_I \eta$ iff there is a computation from (p_0, v_0) to (s, v) in $M_{A,\eta,I}$, for some v. Furthermore, $|S| = O(|Q| \cdot |X|^{|X|} \cdot (max\{C_{A:\eta}, ||I||\} + 2)^{|X|})$.*

Consider the *RSC* problem for an m-dimensional TPVASS A (with respect to predicate η). From A we construct a new TPVASS of dimension $m+1$ in such a way that from each configuration satisfying η, a transition incrementing the new (i.e., $(m+1)$-th) position by one is introduced. Clearly the *RSC* problem has a solution iff in the new TPVASS, there is a computation reaching a configuration with non-zero in the $(m+1)$-th position. Based upon the discussion above and the fact that an m-dimensional VASS A (with n and l as the number of states and the largest integer mentioned in A, respectively) can be simulated by an $(m+3)$-dimensional VAS A' whose largest integer is bounded by $max\{n^2, l\}$ (from Lemma 2.1, [9]), we have:

COROLLARY 9 *Given an m-dimensional TPVASS A, an interpretation I, and an upper-bound predicate $\eta \in B(Q, X, H)$, we can construct an $(m+4)$-dimensional VAS $W_{A,\eta} = (v'_0, V')$ such that $A \leadsto_I \eta$ iff there is a computation from v'_0 to a vector $v'' \geq (0, ..., 0, 1)$ in the VAS, for some v''. Furthermore,*

$||W_{A,\eta}||$ *is bounded by* $max\{(2|Q|\cdot|X|^{|X|}\cdot(max\{C_{A:\eta},||I||\}+2)^{|X|})^2, ||V||, ||v_0||\}$.

It is known from [11] that given a VAS $W = (v_0, V)$, a vector v can be covered in W iff there exists a short covering path whose length is bounded by $O(2^{2^{cnlogn}})$, where c is a constant and n is the size of the VAS. By treating the *dimension* and *norm* of a VAS as two separate parameters, an improved bound of $O(s^{2^{d*m*logm}})$ for the length of the shortest covering path (where s and m are the norm and dimension of the VAS, respectively, and d is a constant) can be found in [12]. This, in conjunction with Corollary 9, allows us to derive the following result.

THEOREM 10 *Given an upper-bound TPVASS A and an upper-bound predicate* η, $||min(RSC(A,\eta))||$ *is bounded by* $O((|Q|\cdot|X|\cdot C_{A:\eta})^{2^{c\cdot m\cdot logm}\cdot|X|^{d|H|}})$, *where* c, d *are constants.*

In what follows, we propose a framework using which the sizes of the minimal elements in an upward closed set can be calculated. The idea is the following. In [13], the key in proving decidability lies in the ability of, given an arbitrary $v \in N^k_\infty$, testing whether '$reg(v) \cap U \neq \emptyset$?'. Now suppose in addition to the ability to test '$reg(v) \cap U \neq \emptyset$?', we are also able to compute the size of a witnessing vector w in $reg(v) \cap U$, if such a vector exists. That is, the small witness property holds for the system under consideration. In this case, the following result can be proven along a line similar to the proof of Theorem 1 presented in [13]. More precisely,

THEOREM 11 *For each upward-closed set* $U(\subseteq N^k)$, *if given a* $v \in N^k_\infty$, *a witness* w *for '*$reg(v) \cap U \neq \emptyset$*' (if one exists) can be computed such that* $||w|| \leq f(||v||)$, *for some function* f, *then* $||min(U)|| \leq (\overbrace{f \circ \cdots \circ f}^{k})(1)$.

4 Computing Maximal Elements for TPAs with Downward-Closed Solution Spaces

For TPAs with unknown lower-bound timing parameters, we still want to find characterization for the solution interpretation. In this case, there is one thing worth noting: *The solution space for the unknowns is downward-closed.* Geometrically, this means that the solution space is a union of "bottom-up" funnels. For convenience, we call such bottom-up funnels *cones*, which can be characterized by the maximal solutions of those cones. If we can find the upper-bounds for the maximal solutions, if any, of those cones, then we can shape the solution space in this case.

LEMMA 12 *Given a lower-bound TPA A and a lower-bound predicate η, if $A \leadsto_I \eta$ for some I such that there is an $\underline{\theta}$ with $I(\underline{\theta}) > \Gamma_{A:\eta}$, then for all I' such that I' agrees with I on all parameters except $I'(\underline{\theta}) > I(\underline{\theta})$, $A \leadsto_{I'} \eta$.*

PROOF. If $I(\underline{\theta}) > \Gamma_{A:\eta}$, then there is a run along which a clock x is incremented beyond $\Gamma_{A:\eta}$ and tested against condition $x \sim I(\underline{\theta})$ where $\sim\in \{“>”, “\geq”\}$. This means that there is a path in the region graph of A^{I^M} whose length of time is greater than $\Gamma_{A:\eta}$ and at the end of the path, $x \sim I(\underline{\theta})$ is tested. Let X' be the set of clocks whose reading is greater than $\Gamma_{A:\eta}$ at the end of the path. Since the length of time is $> \Gamma_{A:\eta}$, we can pick a sequence of $\Gamma_{A:\eta} + 1$ regions along the path such that two successive regions are separated by one time unit. Among these regions, there must be two identical regions and between these two regions, the time-length is no less than 1. By repeating this cycle any number of times, we get a new path (or run) ρ' along which the readings of some clocks, including all clocks in X', becomes arbitrarily large when tested against lower-bound conditions with unknowns. This ρ' is not only a run of A^{I^M} bus also a run of $A^{I'}$. This means no matter how big $I'(\underline{\theta})$ is, we can still find a path to make a run in A^{I^M} to make η^{I^M} reachable. Thus the lemma is proven. ∎

Lemma 12 implies that to search for maximal solutions in cones, we only have to check the reachability of $A \leadsto_I \eta$ with $I(\theta) \leq \Gamma_{A:\eta} + 1$ for all θ. If I is a solution, then so is the cone characterized by $\bigwedge_{I(\theta)\leq\Gamma_{A:\eta}} \theta \leq I(\theta)$, which puts no restrictions on those parameters θ with $I(\theta) = \Gamma_{A:\eta} + 1$. Thus a simple way to formulate the algorithm for the RSC problem of lower-bound TPAs is with the following formula for the corresponding solution space:

$$\bigvee_{I:A\leadsto_I\eta\wedge\forall\theta,0\leq I(\theta)\leq\Gamma_{A:\eta}+1} \bigwedge_{I(\theta)\leq\Gamma_{A:\eta}} \theta \leq I(\theta) \tag{3}$$

THEOREM 13 *Given a lower-bound TPA A and a lower-bound predicate η, the size of $||max(A, \eta)||$ can be computed in PSPACE.*

PROOF. In Expression (3), we need PSPACE to query each question of $A \leadsto_I \eta$ given that the constants used in A^I is no greater than $\Gamma_{A:\eta} + 1$. To see this, we know that each such basic query needs the search of a region graph with at most $2|Q| \cdot |X|^{|X|} \cdot (\Gamma_{A:\eta} + 2)^{|X|}$ regions. Expanding the expression in complexity notation, we get

$$\begin{aligned} & O(2|Q| \cdot |X|^{|X|} \cdot (2|Q| \cdot |X|^{|X|} \cdot (C_{A:\eta} + 2)^{|X|} + 2)^{|X|}) \\ = {} & O(2|Q| \cdot |X|^{|X|} \cdot (2^{|X|}|Q|^{|X|} \cdot |X|^{|X|^2} \cdot (C_{A:\eta} + 2)^{|X|^2})) \\ = {} & O\left(2^{|X|+1}|Q|^{|X|+1} \cdot |X|^{|X|^2+|X|} \cdot (C_{A:\eta} + 2)^{|X|^2}\right) \end{aligned}$$

With a counter, we can explore the region graphs in full. The number of bits in this counter is

$$\begin{aligned} & O\left(\log\left(2^{|X|+1}|Q|^{|X|+1}\cdot|X|^{|X|^2+|X|}\cdot(C_{A:\eta}+2)^{|X|^2}\right)\right) \\ = {} & O(|X|+1+(|X|+1)\log|Q|+(|X|^2+|X|)\log|X|+|X|^2\log(C_{A:\eta}+2)) \end{aligned}$$

As can be seen from the above, we need only polynomial space in the counters and polynomial space to record a single region in order to explore the region graph.

Then according to Formula (3), we need $|H|$ counters of $\log(\Gamma_{A:\eta}+1)$ bits each to carry out the enumeration of the outer disjunction. Thus the total memory used is still in PSPACE. ∎

For lower-bound TPVASSs, the argument used in the proof of Lemma 12 does not work, since a 'loop' in the region graph may not be repeatable due to the possibility of a loss in the counter value.

So far we have seen that for restricted subclasses such as upper-bound and lower-bound TPAs, their solution spaces are upward-closed and downward-closed, respectively, and hence semilinear. This, together with a recent result of [10] showing the emptiness problem to be decidable for bipartite TPAs, leaves us to wonder whether the solution space of a bipartite TPA remains semilinear or not. Following a result in [5] that the solution spaces for general TPAs are not necessarily semilinear, it is reasonably easy to show that the solution spaces of bipartite TPAs are not semilinear in general.

5 Conclusion

We have studied in detail the sizes of the minimal (maximal, resp.) elements of upward-closed (downward-closed, resp.) solution spaces associated with upper-bound (lower-bound, resp.) TPAs. A line of future research for upper-bound TPAs (and TPVASSs) is to explore the possibility of manipulating and characterizing the computations and the solution spaces in a symbolic fashion. Earlier work involving symbolic approaches of reasoning about parametric systems includes [6,10]. Finding how tight our complexity bounds for upper-bound and lower-bound TPAs are remains a question to be answered.

Acknowledgments: The authors thank the anonymous referees for their comments and suggestions, which greatly improved the correctness as well as the presentation of this paper.

References

[1] P. Abdulla, A. Annichini, and A. Bouajjani, Symbolic Verification of Lossy Channel Systems: Application to the Bounded Retransmission Protocol, in *Proc. TACAS'99*, LNCS 1579, pp. 208–222, 1999.

[2] R. Alur, C. Courcoubetis, and D. Dill, Model-Checking in Dense Real-Time, *Information and Computation* 104(1), 2-34, 1990.

[3] R. Alur, and D. Dill, Automata for Modeling Real-Time Systems, in *Proc. 17th ICALP*, LNCS 443, pp. 332–335, 1990.

[4] R. Alur, K. Etessami, S. La Torre, and D. Peled, Parametric Temporal Logic for Model Measuring, in *Proc. 26th ICALP*, LNCS 1644, pp. 169-178, 1999.

[5] R. Alur, T. Henzinger, M. Vardi, Parametric Real-Time Reasoning, in *Proc. 25th ACM STOC*, pp. 592–601, 1993.

[6] A. Annichini, E. Asarin, and A. Bouajjani, Symbolic Techniques for Parametric Reasoning about Counter and Clock Systems, in *Proc. 12th CAV*, LNCS 1855, pp 419-449, 2000.

[7] P. Bouyer, Untameable Timed Automata!, in *Proc. STACS 2003*, LNCS 2607, pp. 620-631, 2003.

[8] E. A. Emerson, and R. Trefler, Parametric Quantitative Temporal Reasoning, in *Proc. IEEE LICS*, pp. 336–343, 1999.

[9] J. Hopcroft, and J. Pansiot, On the Reachability Problem for 5-Dimensional Vector Addition Systems, *Theoret. Comput. Sci.*, 8, 135-159, 1979.

[10] T. Hune, J. Romijn, M. Stoekinga, and F. Vaandrager, Linear Parametric Model Checking of Timed Automata, in *Proc. TACAS*, LNCS 2031, pp. 189-203, 2001.

[11] C. Rackoff, The Covering and Boundedness Problems for Vector Addition Systems, *Theoret. Comput. Sci.*, 6, 223-231, 1978.

[12] L. Rosier, and H. Yen, A Multiparameter Analysis of the Boundedness Problem for Vector Addition Systems, *J. Comput. System Sci.*, 32, 105-135, 1986.

[13] R. Valk, and M. Jantzen, The Residue of Vector Sets with Applications to Decidability in Petri Nets, *Acta Informatica*, 21, 643-674, 1985.

[14] F. Wang, Parametric Timing Analysis for Real-Time Systems, *Information and Computation*, 130(2), 131-150, 1996. Also in *Proc. 10th IEEE LICS*, 1995.

[15] F. Wang, Parametric Analysis of Computer Systems, *Formal Methods in System Design*, 17, 39-60, 2000.

[16] F. Wang, and H. Yen, Parametric Optimization of Open Real-Time Systems, in *Proc. SAS 2001*, LNCS 2126, pp. 299-318, 2001.